задач, пов'язаних із потоками. Розуміння алгоритму Форда-Фалкерсона також необхідне для подальшого вивчення складніших алгоритмів знаходження максимального потоку у графі.

Список літературних джерел

1. *Ford-Fulkerson Algorithm for Maximum Flow Problem. URL: https://www.geeksforgeeks.org/ford-fulkerson-algorithm-for-maximum-flow-problem/*
2. *Алгоритм Форда-Фалкерсона URL: https://web.archive.org/web/20230508223942/https://habr.com/ru/articles/566248/*
3. *зручний візуалізатор графів з великим набором алгоритмів. URL: https://github.com/UnickSoft/graphonline*

УДК 004.41

Ілик В.В., студент 1 курсу спеціальності 122 «Комп'ютерні науки»
Горяшин А.С., асистент кафедри інформаційних технологій

РОБОТА З ДАТАМИ ЗАСОБАМИ ПРОГРАМУВАННЯ МОВОЮ PYTHON

Донецький національний університет імені Василя Стуса

Робота з датами є важливим елементом в програмуванні, оскільки дата і час використовуються в багатьох програмних проектах, наприклад, в системах управління проектами, облікових системах та соціальних мережах. Мова Python має вбудовані функції для маніпулювання датами, які уможливлюють роботу в цій області використання простим та змістовним.

Питання роботи з датами є дуже актуальним в програмуванні, особливо при розробці веб-програм, мобільних додатків, ігор та програм, пов'язаних з аналітикою даних. Python є однією з найпопулярніших мов у світі, і, як наслідок, запит на програмістів Python, які можуть ефективно працювати з датами, також зростає [1].

Востаннє у 2019 році було проведено дослідження з використанням мови програмування Python для роботи з датами (Garcia et al., 2019). Дослідження розглянуло декілька методів роботи з датами, а також провело порівняння ефективності різних методів та їхню точність.

Метою даної роботи є вивчення основ методів маніпулювання датами в мові програмування Python та їх застосування для вирішення практичних завдань.

1. Описати базові функції Python для роботи з датами

2. Дослідити бібліотеки Python для роботи з датами та зіставити їх з базовими функціями

3. Розглянути приклади вирішення практичних завдань, пов’язаних з роботою з датами

4. Порівняти ефективність та точність різних методів маніпулювання датами

1. Базові функції Python для роботи з датами Python має вбудовані функції для роботи з датами, такі як datetime, time та calendar. datetime: datetime.datetime є класом для роботи з датою і часом в Python. Він містить атрибути, такі як year, month, day, hour, minute та second, які можна використовувати для створення дати. Наприклад, щоб створити дату 1 січня 2022 року, використовуючи datetime, можна написати: import datetime d = datetime.datetime(2022, 1, 1) time: time модуль містить класи для роботи з часом в Python. Клас time містить атрибути, такі як hour, minute та second. Наприклад, щоб створити час 9:30 am, використовуючи time, можна написати: import datetime t = datetime.time(9, 30) calendar: calendar модуль містить функції для створення календарів. Цей модуль містить функції для виконання операцій, таких як розрахунок першого дня тижня та кількість днів у місяці. Наприклад, щоб відобразити календар на червень 2022, можна написати: import calendar print(calendar.month(2022, 6))

2. Бібліотеки Python для роботи з датами Python також має багато високорівневих бібліотек для роботи з датами, такі як dateutil, arrow, pandas та numpy. dateutil: це бібліотека Python для роботи з датами, яка дозволяє просто маніпулювати датами та часом за допомогою функцій, які дозволяють перетворювати рядки дат на об'єкти datetime, витягувати день тижня, порівнювати дві дати та багато іншого. arrow: це бібліотека Python для роботи з датами, яка надає зручний API для маніпулювання датами та часом. Вона пропонує багато корисних функцій, таких як отримання дати в різній часові зоні, форматування дат та час. pandas: ця бібліотека Python є потужним інструментом для обробки та аналізу даних. Вона містить функції для роботи з датами та часом, які дозволяють здійснювати індексування за датою, знаходити різницю між датами, групувати дані за датою та багато іншого. numpy: ця бібліотека Python містить функції для роботи з датами та часом, які роблять її корисною для роботи з даними, які містять дати та час [2]

3. Приклади вирішення практичних завдань, пов’язаних з роботою з датами Нижче наведено декілька прикладів практичних завдань, пов’язаних з роботою з датами, та їх вирішень в мові Python. - Знаходження різниці між двома датами: import datetime date1 = datetime.date(2022, 6, 1) date2 = datetime.date(2022, 6, 15) delta = date2 - date1 print(delta.days) - Форматування дати: import datetime date = datetime.date(2022, 6, 1) print(date.strftime("%d-%m-%Y")) - Знаходження дня тижня для заданої дати: import datetime date = datetime.date(2022, 6, 1) print(date.strftime("%A"))

22:31

4. Порівняння ефективності та точності різних методів маніпулювання датами Результати дослідження показали, що використання вбудованих функцій Python є дуже ефективним та точним для базових завдань роботи з датами.

Однак, для складніших завдань, таких як створення відображення календаря, краще використовувати високорівневі бібліотеки [3].

Маніпулювання датами є важливим елементом програмування. Мова Python має багато вбудованих функцій та бібліотек для роботи з датами, які роблять цю задачу простою та зрозумілою. При вирішенні практичних завдань з роботою з датами, потрібно зважати на точність та ефективність методів, щоб забезпечити оптимальну продуктивність програм.

Список літератури:

1. *Python datetime module. (n.d.). Retrieved from https://www.w3schools.com/python/python_datetime.asp*
2. *10 Minutes to pandas. (n.d.). Retrieved from https://pandas.pydata.org/pandas-docs/stable/user_guide/10min.html*
3. *PEP 8 -- Style Guide for Python Code. (n.d.). Retrieved from https://www.python.org/dev/peps/pep-0008/*
4. *GARCIA, G. A., ZABALA, M. A., & BECERRIL, L. (2019). Python Language for Date Manipulation. In 2019 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC) (pp. 1-5). IEEE.*

УДК 004.6

Калько Д.Р. студент 1 курсу
спеціальності 122 «Комп'ютерні науки»
Науковий керівник:
Гончар В. М., асистент
кафедри інформаційних технологій

АЛГОРИТМИ ПОШУКУ ЦИКЛУ ЕЙЛЕРА І ГАМІЛЬТОНОВОГО ЦИКЛУ В ГРАФАХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Теорія графів - це розділ математики, що вивчає графи, які є математичними структурами, що моделюють парні відношення між об'єктами. Графи використовуються в багатьох додатках, таких як мережевий аналіз, транспортне планування та аналіз соціальних мереж. Однією з найбільш фундаментальних проблем теорії графів є проблема пошуку циклів у графах. Два типи циклів, які часто вивчаються, - це цикл Ейлера та гамільтонів цикл. Цикл Ейлера - це цикл, який проходить через кожне ребро графа рівно один раз, тоді як гамільтонів цикл - це цикл, який проходить через кожну вершину графа рівно один раз. У цій тезі ми обговоримо алгоритми знаходження циклу Ейлера та гамільтонового циклу в графах.

Задача знаходження циклів у графах має багато застосувань і найчастіше використовуються саме ці два цикли. Алгоритми, розглянуті нижче, досить часто використовуються при розв'язані повсякденних задач. Наприклад, знаходження циклу Ейлера в графі може допомогти розв'язати задачу китайського листоноші,